

Java™ magazine

By and for the Java community 

//SEPTEMBER/OCTOBER 2014 /

WHAT WILL YOU BUILD TODAY?

05
2014 DUKE'S
CHOICE
AWARDS

11
THE MAKER
MOVEMENT
EVOLVES

16
IOT DEVELOPER
CHALLENGE
WINNERS

05 2014 DUKE'S CHOICE AWARDS

We recognize innovative uses of Java technology.



11
**THE MAKER
MOVEMENT
EVOLVES**
Read our interview with
Maker Faire cocreator
Dale Dougherty.

16
**IOT DEVELOPER
CHALLENGE**
Meet the winners of the
first-ever IoT Developer
Challenge.

COMMUNITY

03
From the Editor

16
Java Nation
News, people, books, and events

25
JCP Executive Series
Rise of the Machines
ARM's Zach Shelby on the Internet of
Things, the Maker Movement, and
the JCP

JAVA TECH

32
New to Java
**Fun and Games with
Greenfoot and Kinect**
Create a game that is controlled by
the movement of the players' bodies.

36
Java Architect
**Understanding Java JIT
Compilation with JITWatch**
Learn how to examine JIT logs to see
what the JIT compiler is doing (and why).

43
Java Architect
**Java Apps on iOS and Android—
Now a Reality**
Build mobile apps with Oracle Mobile
Application Framework, a Java-based
hybrid mobile platform.

48
Enterprise Java
**Get Your Java Applications
Enterprise-Ready with
Oracle WebLogic Server**
Make your applications more robust,
secure, highly available, and scalable.

55
Rich Client
**Location-Based Software
Using JavaFX**
Discover the benefits of using
JavaFX for visualization software.

60
Embedded
**Getting Started with
Home Automation**
Leverage openHAB and the huge Java
ecosystem to create a truly smart home.

66
Fix This
Take our streams code challenge!

EDITORIAL

Editor in Chief

[Caroline Kvitka](#)

Community Editor

[Yolande Poirier](#)

Java in Action Editor

[Michelle Kovac](#)

Technology Editor

[Tori Wieldt](#)

Contributing Writer

[Kevin Farnham](#)

Contributing Editors

[Claire Breen](#), [Blair Campbell](#),
[Kay Keppler](#), [Karen Perkins](#)

DESIGN

Senior Creative Director

[Francisco G Delgadillo](#)

Senior Design Director

[Suemi Lam](#)

Design Director

[Richard Merchán](#)

Contributing Designers

[Jaime Ferrand](#), [Diane Murray](#),
[Arianna Pucherelli](#)

Production Designers

[Sheila Brennan](#), [Kathy Cygnarowicz](#)

PUBLISHING

Publisher

[Jennifer Hamilton](#) +1.650.506.3794

Associate Publisher and Audience Development Director

[Karin Kinnear](#) +1.650.506.1985

ADVERTISING SALES

President, Sprocket Media

[Kyle Walkenhorst](#) +1.323.340.8585

Western and Central US, LAD, and Canada, Sprocket Media

[Tom Cometa](#) +1.510.339.2403

Eastern US and EMEA/APAC, Sprocket Media

[Mark Makinney](#) +1.805.709.4745

Advertising Sales Assistant

[Cindy Elhaj](#) +1.626.396.9400 x 201

Mailing-List Rentals

Contact your sales representative.

RESOURCES

Oracle Products

+1.800.367.8674 (US/Canada)

Oracle Services

+1.888.283.0591 (US)

Oracle Press Books

[oraclepressbooks.com](#)

COMMUNITY

JAVA IN ACTION

JAVA TECH

ABOUT US



ARTICLE SUBMISSION

If you are interested in submitting an article, please [e-mail the editors](#).

SUBSCRIPTION INFORMATION

Subscriptions are complimentary for qualified individuals who complete the [subscription form](#).

MAGAZINE CUSTOMER SERVICE

[java@halldata.com](#) **Phone** +1.847.763.9635

PRIVACY

Oracle Publishing allows sharing of its mailing list with selected third parties. If you prefer that your mailing address or e-mail address not be included in this program, contact [Customer Service](#).

Copyright © 2014, Oracle and/or its affiliates. All Rights Reserved. No part of this publication may be reprinted or otherwise reproduced without permission from the editors. JAVA MAGAZINE IS PROVIDED ON AN "AS IS" BASIS. ORACLE EXPRESSLY DISCLAIMS ALL WARRANTIES, WHETHER EXPRESS OR IMPLIED. IN NO EVENT SHALL ORACLE BE LIABLE FOR ANY DAMAGES OF ANY KIND ARISING FROM YOUR USE OF OR RELIANCE ON ANY INFORMATION PROVIDED HEREIN. The information is intended to outline our general product direction. It is intended for information purposes only, and may not be incorporated into any contract. It is not a commitment to deliver any material, code, or functionality, and should not be relied upon in making purchasing decisions. The development, release, and timing of any features or functionality described for Oracle's products remains at the sole discretion of Oracle. Oracle and Java are registered trademarks of Oracle Corporation and/or its affiliates. Other names may be trademarks of their respective owners.

Java Magazine is published bimonthly with a free subscription price by Oracle, 500 Oracle Parkway, MS OPL-3C, Redwood City, CA 94065-1600.

Digital Publishing by [GTxcel](#)



SEPTEMBER 28 – OCTOBER 2, 2014
SAN FRANCISCO

[oracle.com/javaone](#)

REGISTER NOW

Respond Early, Save \$400*

Discover All Things Java

- Learn all about Java 8
- 500+ educational sessions
- 100+ product demos
- Meet with 500+ Java experts



Diamond Sponsor



Silver Sponsor



Bronze Sponsors



*Save \$400 over onsite registration. Visit [oracle.com/javaone/register](#) for early bird registration dates and details.

Copyright © 2014, Oracle and/or its affiliates. All rights reserved. Oracle and Java are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

ORACLE®

//from the editor /

B

ecause we get to build stuff." That was my nine-year-old son's response to my question about why aviation camp at our local airport was his favorite camp this summer. While kids seem to always be building stuff—from sand castles to LEGOs—as adults we have to try harder to keep that desire to build alive. Why build when you can buy? Luckily, the reasons to build, create, and do it yourself are endless. In this issue, we celebrate the spirit of making and the people who are creating new and innovative technologies and services.

In our [interview with Dale Dougherty](#), *Make*: magazine founder and Maker Faire cocreator, we explore what's behind the growing Maker Movement and how it relates to software development. "We're on the cusp of a new creative industry that is taking product design and the web and beginning to link them," he says. "One of the great starting points for developers is a standardized board such as the Arduino."

We also recognize the winners of the [2014 Duke's Choice Awards](#), which honor compelling and innovative uses of Java technology. These winning makers help ensure safer travel, feed hungry refugees, improve crop yields, provide cool technology, and aim to end a decade-long debate in the Java community.

We're also excited to recognize the [winners of the first-ever IoT Developer Challenge](#), sponsored by Oracle Technology Network and Oracle Academy. Winners used embedded Java with computer boards, such as the Raspberry Pi, and other devices and IoT technologies to showcase innovations in domotics, robotics, office design, authentication, and rating systems.

Congratulations to all of the winners, who show us that the spirit of making is alive and well. Inspired? Get out there and build something.

Caroline Kvitka, Editor in Chief 



Find the Most Qualified Java Professionals for your Company's Future

Introducing the *Java Magazine* Career Opportunities section – the ultimate technology recruitment resource.

Place your advertisement and gain immediate access to our audience of top IT professionals worldwide including: corporate and independent developers, IT managers, architects and product managers.

For more information or to place your recruitment ad or listing contact:
tom.cometa@oracle.com

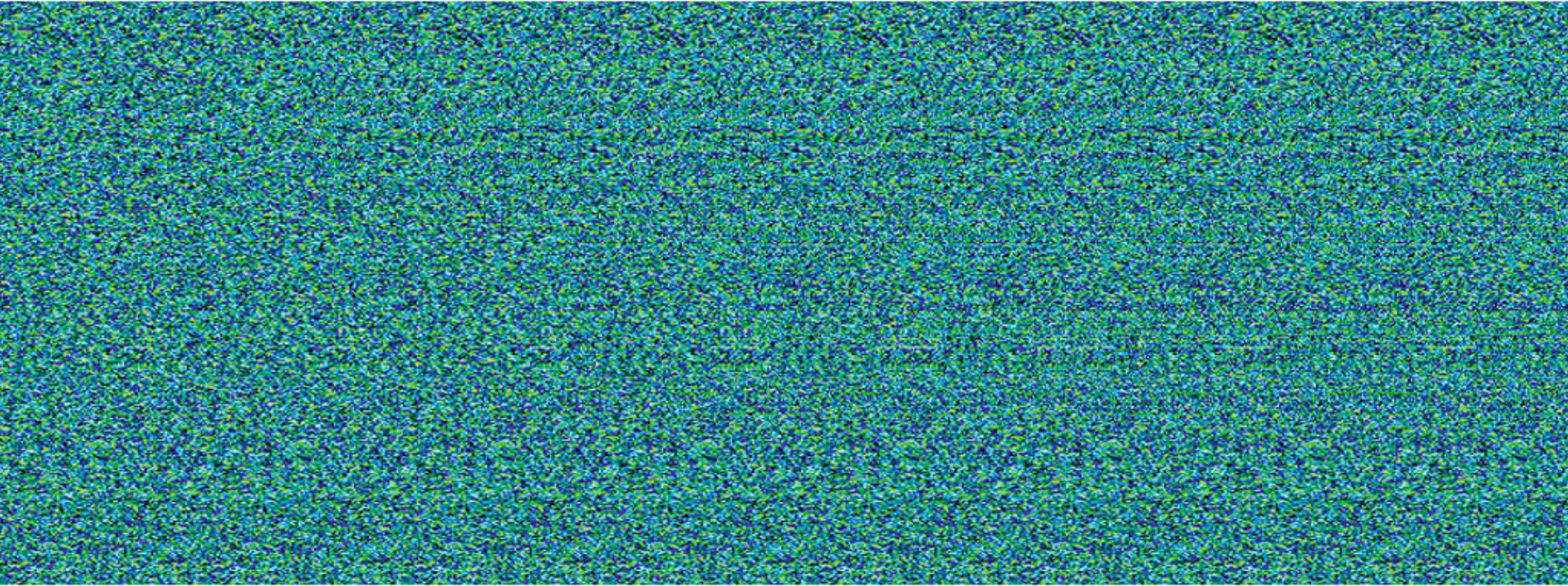
//send us your feedback /

We'll review all suggestions for future improvements. Depending on volume, some messages might not get a direct reply.



PHOTOGRAPH BY BOB ADLER

The answer is right in front of you



Java Image Enabling SDKs that Help You See the Big Picture

At first glance it may seem difficult, but it's really quite simple. Atalasoft's JoltImage product is a proven SDK for image enabling your Java-based web applications, easily. Image enabling helps to add dimension to your data, so you can uncover insights such as correlations and causations hidden inside your 2-dimensional documents. Our SDK does the heavy lifting for you, saving time, money, and the headaches of figuring it out yourself. Backed by our highly knowledgeable & caffeinated support engineers, JoltImage will enable your success and make the big picture so much easier to see.



[Click for tips on viewing the stereogram](#)

Image enabling experts & bacon connoisseurs. Visit us online to see our full line of SDK products for .NET and Java



COMMUNITY

JAVA IN ACTION

JAVA TECH

ABOUT US



05

WFP Subsidy Card,
e-finance for the
United Nations
High Commissioner
for Refugees and
the World Food
Programme

ORACLE.COM/JAVAMAGAZINE ////////////////////////////////// SEPTEMBER/OCTOBER 2014



SAFETY FIRST

The driverless car is still a few years away, but crash avoidance systems are already available in some of today's popular luxury automobiles. Working with the [Insurance Institute for Highway Safety](#) (IIHS), Perrone Robotics has developed the Java-based **Autonomous Vehicle Test System** (AVTS) to independently test these safety systems.

JAVA WINS OUT
"Java took the highest score against other platforms."

—Omer Faruk Arar,
Chief Researcher,
Air Traffic Controller
Selection System

"Many of today's vehicles already have 'crash-imminent braking' and other features," says Paul Perrone, CEO of his eponymous Charlottesville, Virginia-based firm. "But up until now, the IIHS hasn't really had a way to test how well these technologies work in real-world situations with cars traveling at highway speeds."

Perrone Robotics began work on the AVTS in early 2013 and has completed phase one of its contract—which required delivery of one complete system, including one target robot vehicle and one drop-in actuator kit, for use by the IIHS. "They are now ramping up on using the system," says Perrone.

The TUBITAK BILGEM Informatics

and Information Security Research Center, a Turkish government agency, and Air Navigation Service Provider and Airport Operator of Turkey (DHMI) used Java to develop a new computer-based testing system to help select and vet air traffic controllers: the **Air Traffic Controller Selection System** (atcSES). Before atcSES, another computer-based selection system was used, explains atcSES Chief Researcher Omer Faruk Arar. "This system obviously had some drawbacks," says Arar. "It could not provide enough administrative flexibility. With atcSES, necessary flexibility is provided to authorized users and the skills needed for an air traffic controller could be evaluated more interactively by new tests."

With atcSES, the agency evaluates potential candidates on eight essential skills: spatial awareness, reasoning, complex attention, psychomotor abilities, visual memory, auditory memory, cross-control, and basic mathematics.

Before starting the atcSES project, the agency evaluated suitable technologies for the project. "Java took the highest score against other platforms," says Arar. "We needed to implement tests quickly and adapt the system easily. By using Java, we easily achieved that."

Left: Paul Perrone at the Insurance Institute for Highway Safety. Right: Perrone and team install and test their Autonomous Vehicle Test System.



PHOTOGRAPHS BY PAT JARRETT/GETTY IMAGES



Tomas "Tito" Sanchez (right) of the International Maize and Wheat Improvement Center analyzes crops with IBFieldbook on a tablet computer.



FIGHTING HUNGER

Two of this year's winning projects are helping fight hunger. The first is the **WFP Subsidy Card**, which is based on Java smartcard technology and was developed by [e-finance](#) for the United Nations High Commissioner for Refugees (UNHCR) and the World Food Programme (WFP).

"The WFP Subsidy Card helps refugees listed by the United Nations get the food they need through secure-card wallets controlled by both organizations," explains Mohamed Taman, systems architect and project manager for Cairo, Egypt-based e-finance. "Right

now they're being used by 100,000 refugees; by the end of the year they will be helping 500,000."

This is the second Duke's Choice Award for a UNHCR project; the U.N. global refugee agency won in 2012 for Level One, a light client application that enables U.N. personnel to register refugees and assess their immediate needs.

In Texcoco, Mexico, the International Maize and Wheat Improvement Center—el Centro Internacional de Mejoramiento de Maíz y Trigo (CIMMYT)—has used Java, JavaFX, and NetBeans to create a crop and field analysis tool to improve crop quality and yields around the world: Integrated

Breeding Field Book, or **IBFieldbook**. This application helps design field trials and generate crop analysis and enhancement solutions. In conjunction with tablet computers in the field, it efficiently and accurately captures crop phenotype data and saves it in a local crop database before submitting that information to CIMMYT for analysis.

"Before IBFieldbook, people recorded data using Excel spreadsheets and Excel macros before storing the information in Access databases," explains development team member Tomas "Tito" Sanchez. "We also used some tools developed in Delphi and VB to query the database, consuming a lot of time and generating serious bottlenecks to access the information for analyses. The problem was that, due to different Excel versions, the application was not able to run on different operating systems. That's why we decided to use Java and NetBeans—because we will be able to deploy IBFieldbook on Linux, Windows, and the Mac OS."

PHOTOGRAPH BY RAFAEL MONROY/
GETTY IMAGES



COOL TECH

No Duke's Choice Award winners roster would be complete without some "cool" technology. This year the honors go to the **PiDome** home automation and Internet of Things (IoT) platform, which marries Java and the Raspberry Pi credit-card-sized single-board computer. Some important features are real-time USB device recognition; Raspberry Pi hardware interfaces; a visual trigger editor for any controllable device; and a visual floor planner, which is visualized in 3-D in the PiDome JavaFX OS-themed client with real-time updates.

"What we are trying to do is to take all those great home automation projects by electronic hobbyists and fellow Java developers and put them together in one single package. By supporting existing products, open source projects via plugins, and our own hardware created by Marcel Wiebenga, it will be able to compete with the larger home automation systems," says John Sirach, project leader at PiDome.

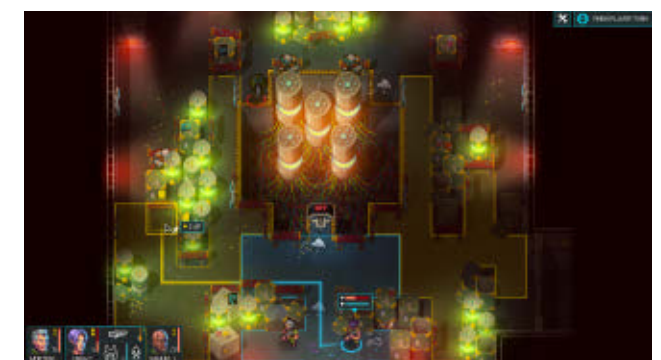
From a diverse team of open source enthusiasts comes **libGDX**, a cross-platform game development framework that allows programmers to



write, test, and debug Java games on a desktop PC running Windows, Linux, or Mac OS X and deploy that same code to Android, iOS, and WebGL-enabled browsers—something not widely available right now. The goal of libGDX, says creator Mario Zechner, "is to fulfill the 'write once, run anywhere' promise of the Java platform specifically for game development."

DukeScript is an alternative to Swing, SWT, and JavaFX, with the benefit that it also runs on Android and iOS and enables true cross-platform Java.

"Although the name somehow seems to imply otherwise, DukeScript is not a scripting language," says DukeScript collaborator Anton (Toni) Eppler of Munich, Germany-based Eppler IT Consulting. "DukeScript applications are plain Java applications that internally



Top: Mario Zechner of libGDX. Center and bottom: screenshots of *Halfway*, a game developed in libGDX.

STRONGER AS ONE

"We are trying to take all those great home automation projects ... and put them together in one single package."

—John Sirach, Project Leader, PiDome

PHOTOGRAPH BY TON HENDRIKS



JCertif Conference attendees in Africa



use HTML5 technologies and JavaScript for rendering. This way developers only need to write clean Java code and can still leverage the latest developments in modern UI technology.” DukeScript is developed as a subproject of NetBeans.

Helping educate future technologists about Java and open source technologies in Francophone Africa is **JCertif International**. Since its founding in 2010, this organization has trained

5,000 Java developers across Africa, according to Founder and Managing Director Max Bonbhel. “JCertif International is an independent, non-profit organization that brings together African community leaders dedicated to growing and empowering developer communities through training and events,” explains Bonbhel. “JCertif also provides free training materials and speakers and promotes collaboration

between tech user groups across the African continent.”

Its JCertif University, held several times a year in various cities across the continent, offers two- to five-day intensive classes and workshops, typically for 20 to 50 students at a time. Its annual JCertif Conference is a one- to two-day conference with lectures and speakers that typically attracts up to 2,000 attendees, says Bonbhel.

Three-Way Tie Makes 12 Winners

How fitting that in its 12th year the Duke's Choice Awards has 12 winners, thanks to the first-ever three-way tie in the voting for the coveted Community Choice Award.

The first of the trio is **Project JEDI**, for the Java Education & Development Initiative. This community-driven organization provides free courseware and training on Java and open source technologies to colleges and universities around the world. Originally started in the Philippines, Project JEDI now partners with Java user groups (JUGs), academic institutions, government agencies, and businesses around the world to translate and provide the course materials in other languages.

The second winner is the Apache Software Foundation's **Apache DeltaSpike** developer toolbox for Java Contexts and Dependency Injection (CDI) for the Java EE 6 platform. Apache DeltaSpike allows developers to introduce and test new features in Java EE before they are standardized, such as transaction support for CDI beans, which wasn't available until Java EE 7.

The third winner, **JavaFXPorts**, from LodgON, is also an open source tool-kit. For the first time, it makes graphically rich Java and JavaFX available to mobile and tablet devices running iOS and Android operating systems.

by the Maker Movement is opening up rich possibilities for Java developers. The recent *MakerCon Conference*, hosted at Oracle headquarters, reinforced that as embedded devices get cheaper, more powerful, and more connected, and as the IoT grows, Java developers will face new challenges and opportunities and will look to Oracle for strong support. To that end, a major effort to unify the Java platform is underway—specifically, Java ME with Java SE—to shrink Java SE into the embedded space for smaller devices.

Dale Dougherty, CEO of Maker Media, founder and publisher of *Make: magazine*, and cochair of MakerCon, is at the center of the Maker Movement. Make:, which Dougherty launched in 2005, provides a wealth of wide-ranging information about do-it-yourself and do-it-with-others projects, along with step-by-step instructions on how to make things.

Dougherty, the cocreator of Maker Faire, has a rich and creative history, much of it tied to O'Reilly Media. Prior to Make:, he developed Global Network Navigator (GNN), the first commercial website, which was launched in 1993 and sold to America Online in 1995. From 1995 to 1999 he was publisher of Web Review, an online magazine for web designers. As publisher of the O'Reilly Network, which he founded with Tim O'Reilly, he developed the Hacks series of books and was honored by the White House as a "Champion of Change." We talked with him to get an insider's view of the Maker Movement.

PHOTOGRAPH BY BLAINE FISHER/GETTY IMAGES



Java Magazine: What's been happening lately in the Maker Movement?

Dougherty: For starters, we had a White House Maker Faire recently, which tells us that the Maker Movement is moving from the margins to the mainstream. I'm quite happy about the impact it can have in areas such as manufacturing, innovation, and especially education—and even in civic and community areas as well. The Maker Movement is a widely distributed social network that anyone can join. You identify yourself as a maker and you're in.

We are trying to reach people who don't identify as makers and show them

that there are pathways to become makers. We believe that there is a set of values that you acquire by making. And one of the key ones is to see yourself as a producer—a shaper and a builder—and not just a consumer. It really doesn't matter whether you're working with wood, plastic, 3-D printing, or flour and egg. Making can take many different forms. We feel that making is important in helping people lead more productive and meaningful lives.

We also see that making is a source of innovation. Tools for people who have ideas for services and things are becoming increasingly available and

Dale Dougherty gives a presentation at Maker Faire Kansas City on June 28, 2014.



Left: Dougherty at Maker Faire Kansas City. Right: Kids learn and play at Maker Faire Bay Area.

easy to use. And the time it takes to make things is collapsing. So you might be scratching your head with a good idea and feel like, well, I don't know how to get this done. Making initially has a DIY nature, but as people get involved they meet others and there is more collaborative production.

I met a man recently who was paralyzed from a diving accident and is confined to a wheelchair that has a vertical pole like a joystick that is used for control. He has a paralyzed hand and can't wrap his hand around the pole to use it effectively—in fact, he was just resting his hand

on the top tip of it. So, using a 3-D program, he designed a mitt interface that enables him to wrap his hand around the stick and navigate his wheelchair. He had it 3-D printed in plastic and then printed in metal. It's empowered him to enhance his mobility.

He represents the core of the Maker Movement. Once the door is opened he starts thinking, "Well, there must be other people like me who need this. So what I designed for myself could help other people."

Java Magazine: The Maker Movement seems to affirm certain core human needs to create,

Java, Meet Maker Movement

We asked Noel Portugal, principal user experience developer at Oracle, to weigh in on Java's role in the Maker Movement.

"Java has become the de facto language for enterprise applications. A lot of libraries and code examples that are commonly used in enterprise applications can now be used in conjunction with the Maker Movement through embedded Java and the Internet of Things [IoT]," says Portugal. "Data collected by sensors needs to be analyzed by microprocessors and then sent to the cloud. This is easily achieved using a Java Virtual Machine [JVM] and the Java language."

Java developers already have the skills to create the IoT and build devices with the potential to change the way we live and interact with the world, he says. "It's up to you to dream up and create a physical device that will connect to the cloud. There are no limits to the imagination."

Portugal adds that the Oracle Java team has worked closely with the maker community to make sure that embedded Java works out of the box with embedded ARM microprocessors. "Performance is key for real-time applications, and embedded Java applications are up to the task. The maker community is always welcoming and ready to help."

People are creating other boards and things that have Wi-Fi baked into them. The Raspberry Pi runs a full Linux environment where you have the tools available for your program source development. The key is, how do you connect with the physical world? You may be

And what happens when lots of different things in your “built” environment are smart and interacting with you and each other? Take an area such as wearables, which might be a watch or clothing. What behaviors do we want associated with them? Someone recently told me that they were interested in having a sweater that was programmed so that it knew that people were looking at it and reacted. It’s a playful idea that gets us leaping into the world of the possible. It gets



Dougherty talks with a young maker at Maker Faire Kansas City.

An architect at a Maker Faire was showing a Tesla coil that arced through the air in sync with music that was playing. We could have whole buildings doing that. On the other hand, there is a side to this that is very practical.



A Maker Faire Kansas City maker shares her story with Dougherty.

A friend of mine told me about a “cat detector” that he built. He and his family had adopted a stray cat that had a lot of scratches and wasn’t doing so well, so they started putting out food for it. They rarely saw the cat, but they saw the empty food bowl and wondered who was eating the food. So he took a little camera and connected it to a motion sensor, along with a trigger light set up by the cat food, so that when something approached the bowl, it took a picture and sent it to his cell phone. They discovered that both the cat and a raccoon were eating the food. It’s a very practical example of solving a specific problem, such as the one for the man in the wheelchair. People now can see

a problem and whip something together off the shelf and solve it. I don’t know if there’s a big market for a “cat detector,” but you would be surprised at the number of similar stories from people who want to know what’s happening in their yard or their chicken coop at night.

Java developers, once they get acquainted with electronics, are in a position to do things on a much grander scale. There’s something a bit disorienting about the Maker Movement, but it’s a kind of creative disorientation because you become aware of so many possibilities. You go to a Maker Faire and see 10 things in a row that are extremely different from each

other, each of which leads you to think of new options. You see someone doing some new kind of crochet and someone else with a robot, and soon you start seeing robots doing crochet. And if you are good at coding, you have a great advantage with things such as the Arduino platform, BeagleBone, the Raspberry Pi, and new areas such as the Spark Core board.

We published a really great book called *Make: Electronics* that can help developers make the journey into electronics. One interesting direction is JSON [JavaScript Object Notation] interfaces, which should enable web developers to talk to sensors in a language they are familiar with.

Java Magazine: Any final remarks?

Dougherty: We hear a lot about the IoT. It’s important to keep in mind the new social and human space that gets created when there is a lot of data being passed between devices. We are talking about new kinds of human experiences and behaviors for everyone. The Maker Movement is raising a flag and saying that everyone has something to contribute in creating a new-and-improved world.

What I’m most thrilled about is the number of kids showing up in the Maker Movement. They are so excited and engaged. I really want to help create a world where kids can find places in their community where they learn to make things and become makers themselves. The key to this is the creation of maker spaces in schools and libraries, museums and community centers, formal and informal settings, so that kids can have access to tools and materials and find mentors. This will not only create new innovators but it will help people gain better control of their lives. **</article>**

Timothy Beneke is a freelance writer and editor who has written for *Mother Jones*, the *East Bay Express*, and the *Chicago Reader*.

LEARN MORE

- [MAKE: magazine](#)
- [White House Maker Faire](#)
- [“A Perfect Match: Java and the Internet of Things”](#)



JVM LANGUAGE SUMMIT



Oracle JVM Architect John Rose discusses summit topics.



Waratek's Nigel Daniels talks about security solutions for the JVM.

Summit highlights (from top to bottom): Participants mingle during a break; summit presenter Charles Nutter; Brian Goetz (center, in yellow) chats with attendees.

The seventh annual Java Virtual Machine (JVM) Language Summit took place July 28–30, 2014, at Oracle's Santa Clara, California, campus. This event was an open technical collaboration of 90 attendees, including language designers, compiler writers, tool builders, runtime engineers, and VM architects. The format, a mix of technical presentations and workshops, facilitated discussion groups and enabled deeper dives into the subject matter.

"I'm enjoying the company of a lot of brilliant colleagues," said **John Rose**, JVM architect at Oracle. "For many of us, the JVM Language Summit is the most enjoyable day of the year, professionally, because we get to exchange ideas with our peers from Oracle and from other parts of the Java ecosystem."

Get slides and more from the summit [here](#).

INTRODUCING ROBObURGH



C.H.I.M.P., the CMU Highly Intelligent Mobile Platform, was showcased at the Launch CMU event at Oracle headquarters.

Technology is changing Pittsburgh's image from the steel city of the 1970s to a twenty-first-century hub for innovation. And much of that is

due to the efforts of the Center for Innovation and Entrepreneurship at Carnegie Mellon University (CMU). One area that is thriving is robotics—so much so that Pittsburgh may have to change its name to “Roboburgh.”

A select group of Roboburgh residents made the trip to another famous hub of innovation, Silicon Valley, to attend the Launch CMU event at Oracle headquarters in Redwood Shores, California, earlier this summer.

As CMU faculty and alumni began to showcase their inventions to possible investors, it quickly became clear that robots are much more than just a box-of-bolts android that can go out to the front lawn and pick up a newspaper. As **Rob Daley**, CEO and cofounder of CMU-incubated startup 4moms, put it, “There is a whole lot of very core fundamental technology that’s related to robotics that’s really powerful and really important if you divorce yourself from the media

concept of humanoid robotics.”

Daley’s company was just one of the many organizations that showed up to explain how they use robotics technology in their products. Representatives from 27 startup companies in industries ranging from healthcare to music were represented at the two demo and poster sessions, which bookended the showcase talks.

Manuela Veloso, a professor of computer science at CMU, said the most-powerful robots are the ones that can ask for help. The “CoBot” robots in Veloso’s lab use Java in an Android app as a GUI to enable humans to speak with the robots, providing an interactive capability Veloso called “symbiotic autonomy.” Java also powers an app that humans can use to summon the robots. “In our work, Java is used to enable robots to interact with people through phone apps or by accessing the web, or to visualize the robots’ state,” said Veloso.

Whether these robots transform Pittsburgh into Roboburgh is yet to be seen. But one thing’s for sure: These aren’t your mother’s robots.

Left: a session at Open Cloud Day.
Right: Cody Herriges of Puppet Labs presents on open cloud APIs.



OSCON: OPEN CLOUD DAY

Open Cloud Day, held July 21, 2014, during the OSCON open source convention, focused on the latest innovations in public and private clouds, infrastructure as a service (IaaS), and platform as a service (PaaS). In his session, "The Enterprise Challenge for Cloud Computing," **David Nalley**, committer to Apache CloudStack, tackled the question, "Why hasn't cloud computing dominated the enterprise already?"

Nalley said that the cloud can be great for certain things, such as application test beds, but thinking that the cloud will solve all your problems is a little like believing in unicorns. He asked the audience to consider value, pay for use, fiefdoms, and fragility when thinking about moving enterprise

applications to the cloud.

The cloud is also forcing enterprises to rethink their IT services. "People are really more empowered at home than at work," explained **Chris Launey**, director of cloud services at Disney, in his session. Launey doesn't want Disney developers to come up with an idea over the weekend, begin developing it using free services they can access at home, and then run into a brick wall when they try to implement it at work on Monday. Things need to be much, much faster. "I challenge my team to do things 'cloudly,'" he said. "That means we need to make requests stupidly easy, provide fast delivery, allow for painless management, and provide data transparency and granular billing."

JCP Celebrates 15 Years



The Java Community Process (JCP) celebrated its 15th anniversary with a gathering at the Computer History Museum in Mountain View,

California, on June 18, 2014. The event, cohosted by the JCP program office and the Silicon Valley Java User Group, was a chance for members of the Java community and some of the people instrumental in the JCP's success to network and check out the cool museum exhibits.

"For 15 years the JCP has been working with corporations, individuals, nonprofits, and Java user groups to advance and improve the Java platform," said JCP Chair **Patrick Curran**. "Every few years we like to get together and say thank you, because Java would be nothing without you."

He recognized many of the luminaries who had played a part in the JCP over the years, including **James Gosling, George Paolini, Rob Gingell, Onno Kluyt, Bill Shannon, and Vineet Gupta**, and then asked everyone to raise a glass to the JCP. Later, Duke shook hands and posed with attendees.

Watch **Stephen Chin's** interviews with [Gosling](#), [Gingell](#), [Van Riper](#) and [Kevin Nilson](#), and [others](#).



DEVOXX UK: MIND THE GEEK

If you're a curious developer, Devovxx UK is an amazing conference.

Held June 12–13, 2014, the conference had more attendees, more sessions, and more sponsors than last year. Java and Java 8 got lots of attention, with several sessions on lambdas and other Java 8 features. **Venkat Subramaniam**'s session, "Programming with Lambda Expressions," was full. He explained that lambdas are good in themselves, but their real value is showing you what's possible with the Stream API. "Lambdas are the gateway drug to streams," he said with a laugh. Experts discussed JavaFX; security; Java EE; and, of course, the Internet of Things (IoT).

In his session, **Dick Wall** showed his IoT Venn diagram and explained that IoT is exploding. He admitted that his dog has a fitness band and is Wi-Fi-enabled. Wall said it's not just the internet anymore; it's the "ubiquinet."

At the closing keynote, London Java Community co-organizer **Martijn Verburg** encouraged attendees to start using Java 8 and to hook it into their continuous integration servers. He also said that every developer is an architect, and if you are an architect who doesn't code regularly, "you should quit right now." If you are a developer who isn't comfortable considering the architecture of applications and working on a whiteboard, he said, "you should consider another profession." Finally, he encouraged developers to become DevOps pros, and to automate everything that they can.

Mark your calendar for June 17–20, 2015, for the next Devovxx UK.

Clockwise from top: a Devovxx UK expert panel, "Fearless Change" presenters Linda van der Pal (left) and Régina ten Bruggencate, a red-hatted attendee with a message





Left to right: ARM's Zach Shelby, Patul Sharma, and Sergio Scaglia walk to a meeting.

Zach Shelby is director of technical marketing for Internet of Things at ARM and a recognized industry thought leader. Shelby cofounded Sensinode, where he served as CEO, CTO, and chief nerd before the company's recent acquisition by ARM. Prior to Sensinode, he led wireless networking research at the Centre for Wireless Communications and at the Technical Research Center of Finland.

Shelby is a key contributor at the Internet Engineering Task Force (IETF), Open Mobile Alliance (OMA), and ZigBee for IoT standards and has contributed to 6LoWPAN, routing, web services, and security-related standards, and to several international research programs. Shelby is known as a pioneer in the use of IP and web technology in low-power networks with 6LoWPAN and Constrained Application Protocol (CoAP) standards development, and is coauthor of the book 6LoWPAN: The Wireless Embedded Internet. His large portfolio includes courses, publications,

JCP Executive Series

Rise of the Machines

ARM's Zach Shelby discusses the Internet of Things (IoT), evolving standards, the Maker Movement, and how the JCP helps to facilitate these dynamic technologies. **BY STEVE MELOAN**

PHOTOGRAPHY BY BOB ADLER

MAKING DATA MEANINGFUL

“For the IoT, we need to build semantics and meaning into data received from devices, going out to cloud services. For instance, what does temperature mean?”

dred KB/sec, shared across tens or even thousands of devices. Without these low-bandwidth networks, we don't really have an IoT; we just have mobile cellular devices and Ethernet-connected industrial controllers, which has been the past paradigm.

6LoWPAN technology is really no different from any other IP networking technology. It can be run on almost any network—ZigBee, Sub-GHz, Bluetooth Low Energy, and so on. It's really just a way of compressing IPv6 over all those different technologies.

A new standard called Thread will simplify implementing 6LoWPAN technology into home devices. The focus is home automation and home security. The goal is to make these implementations easily available and certifiable by the product manufacturers.

From the Java perspective, Thread is just another networking interface. It will look like a Wi-Fi network, or any other network that transmits IP packets.

Java Magazine: How important is the Maker Movement in the IoT domain?

Shelby: The traditional picture of embedded developers has been nerds with horn-rimmed glasses hacking a

test harness and debugging by looking at signals on a wire with an oscilloscope. That's still important work; we need people to create those underlying technologies. But now we need products that are exciting and sexy.

The Maker Movement is creating this kind of energy, facilitating inventors in 3-D printing; robotics; Computer Numerical Control [CNC]; and even woodworking, metal working, and traditional arts and crafts. In response to this movement, developers are producing tools that have broader applicability. The mbed platform is an example of this evolution. The Arduino single-board microcontroller is another example. It has made the experience of configuring an embedded device very simple. We've just launched a new ARM-based Arduino device.

The Maker Movement will motivate more and more technology to service these widely diverse activities. Professional developers, product manufacturers, startups, and hobbyists will all be in the game.

Java Magazine: What are some of the standards evolving in this space? And how does the IPSO Alliance promote innovation?

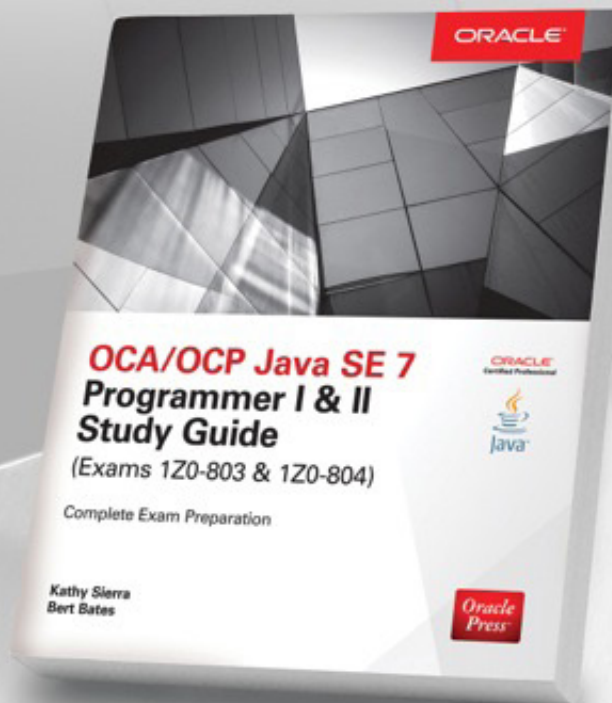


Shelby: For the IoT, we need to build semantics and meaning into data received from devices, going out to cloud services. For instance, what does temperature mean? And what is the location of this device?

In the world of M2M, everyone baked their own semantics into the protocols. And it was created from scratch every time. A satisfying challenge for engi-

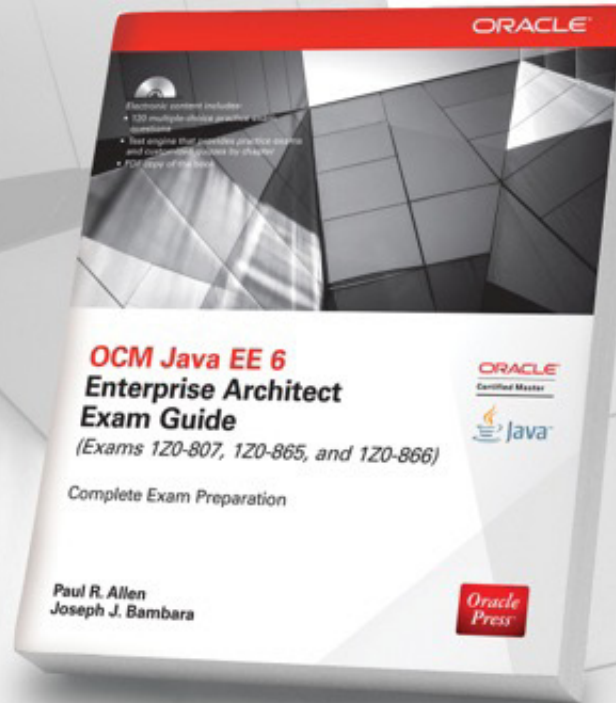
Shelby chats with Dominic Pajak at ARM's San Jose, California, offices.

Written by leading Java experts, Oracle Press books offer the most definitive, complete, and up-to-date coverage of Java available.



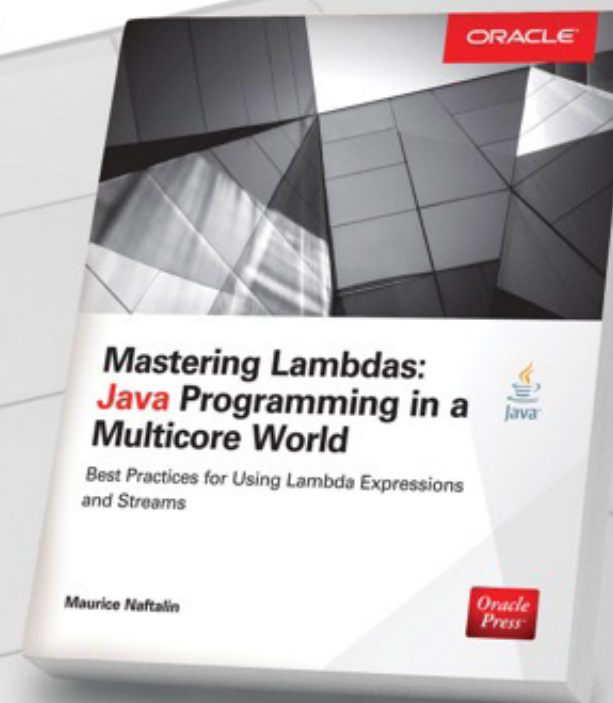
**OCA/OCJP Java SE 7
Programmer I & II Study Guide**
Kathy Sierra, Bert Bates

This self-study tool offers full coverage of OCA/OCJP Exams 1Z0-803 and 1Z0-804. Electronic content includes 500 practice exam questions.



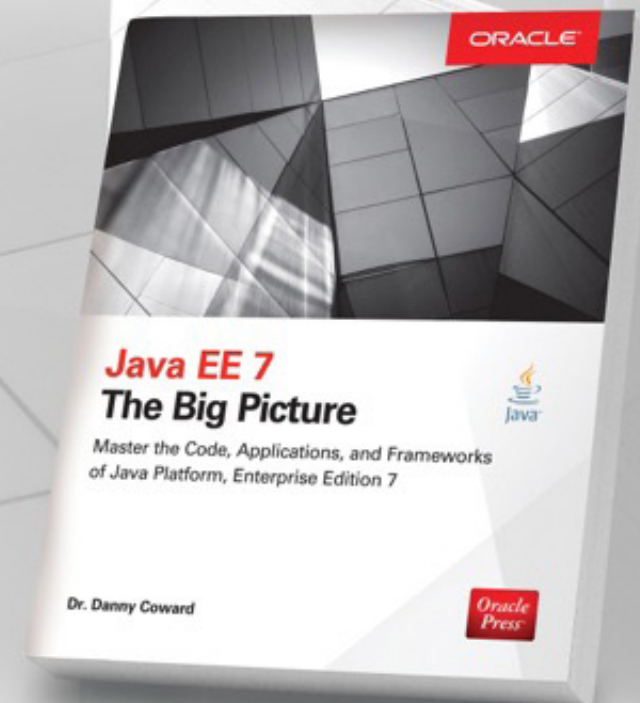
**OCM Java EE 6 Enterprise
Architect Exam Guide**
Paul Allen, Joseph Bambara

Get complete details on Oracle Certified Master Java EE 6 Enterprise Architect Exams 1Z0-807, 1Z0-865, and 1Z0-866. 120 multiple-choice practice exam questions are included.



**Mastering Lambdas
Java Programming in a
Multicore World**
Maurice Naftalin

Effectively use Lambda expressions to maximize performance improvements provided by multicore hardware.



Java EE 7: The Big Picture
Danny Coward

Master the code, applications, and frameworks that power Java Platform, Enterprise Edition 7.

then use that object to get more-detailed information, such as the x-coordinate and y-coordinate of the joint on the screen. (Refer to **Figure 3**.) The `Joint` class also contains constants for identifying each individual joint.

The setup. Our own scenario gets initialized in the `PaintWorld` class. We will not discuss the code in detail here; you can read the code yourself if you are curious, or you can ignore it for now. But here is a summary of its purpose.

The **PaintWorld** object, in every *Greenfoot act cycle*, looks for all *tracked users* (that is, all users who are in the view of the Kinect camera) and creates a **Canvas** object for every user it detects.

A **Canvas** is a Greenfoot object that has a transparent image the size of the entire Greenfoot world, and it is linked to one tracked user. Since it is entirely transparent, it will initially be invisible. So if Kinect currently sees, for instance, two users, there will be two **Canvas** objects in the Greenfoot world acting as (invisible) painting layers, and each user paints

on his or her own layer.

The Canvas class. We will do all our work in the **Canvas** class. This is where all the fun is.

Let's have a quick look at the constructor code shown in **Listing 1**.

We can see that the `UserData` object for the user associated with this canvas is received and stored for later use. Then a transparent Greenfoot image on which to paint is created. Lastly, a random color is selected from an array of predefined colors (defined further up in the class), and it is set as the color for future painting operations.

The act method.

Although the constructor is executed only once, the Greenfoot `act` method is executed repeatedly as long as the scenario is running. It, too, is fairly simple (see **Listing 2**).

In this method, we use the `user` object to get information about the right hand by using the `user.getJoint(Joint.RIGHT_HAND)` method call. This will give us an object of type `Joint`.

We can then use the joint's `getX()` and `getY()` methods to determine the right hand's position on the screen and then

GET PHYSICAL
By connecting Microsoft Kinect to Greenfoot, **we can create games that we can control with our bodies:** No hitting of keys and no mouse clicks—just jumping up and down and waving your arms in the air.

LISTING 1

LISTING 2

LISTING 3

LISTING 4

```
public Canvas(int width, int height, UserData user)
{
    this.user = user;
    setImage(new GreenfootImage(width, height));
    getImage().setColor(
        colors[Greenfoot.getRandomNumber(colors.length)]);
}
```


[Download all listings in this issue as text](#)

to paint a circle where the hand is. This is achieved by using the `fillOval` method with the x-coordinate and y-coordinate, and using 20 as both width and height of the circle.

`fillOval` is a method of the `GreenfootImage` class, which we can call on the canvas's image. We retrieve the image using the `getImage()` method, which is one of the predefined methods of the `GreenfootActor` class.

One last thing that we can see in **Listing 2** is an `if` statement that checks whether the right hand is the joint nearest to the camera. We draw only if it is, so we can move the right hand without painting just by holding it farther away from the screen.

This short code segment shows how easy it is to follow a particular joint and to draw at its location.

Now, let's jump in and write some code of our own.

Task 1: Erasing the Screen

Our first task is to erase the screen when we lift our left hand over our head.

Erasing the screen is very easy, since the `GreenfootImage` class has a `clear()` method. So the call to erase a user's image is simply the following:

```
getImage().clear();
```

All that is left to do is to check whether the left hand was raised above the head. We can do this by retrieving the y-coordinate for both the left hand and the head, and then comparing them, as shown in **Listing 3**.

Now we just need to put these two bits together in an **if** statement, as shown in **Listing 4**.

Try it out. Add this code to the `act` method of your `Canvas` class (after the painting code), compile,

Microsoft Windows, use `launchUI.bat` instead). You'll be presented with the simple welcome screen shown in **Figure 1**.

From here, you can load a JIT compilation log. You might not have one of these on hand, so let's look at a JITWatch script for generating a test log. Before we do that, however, it's worth discussing assembly code and its importance to JITWatch.

Working with Assembly Code

To get the most out of working with JITWatch, we need to look at the machine code generated by the JIT compiler. This is done by using a HotSpot disassembler (HSDIS) plugin to turn the generated code back into human-readable assembly code. This plugin then enables us to use the `PrintAssembly` flag to get assembly code dumps, which means that we can get a great deal more out of JITWatch.

The plugin that is usually used is the [base-hsdis](#) plugin, maintained by John Rose and others. It is available from the Downloads section of this website.

Download the correct plugin for your operating system, and copy it to the directory that contains the Java Virtual Machine (JVM) library file ([libjvm.so](#) on Linux, [libjvm.dylib](#) on Mac, [libjvm.dll](#) on Microsoft Windows). On Java SE 8, this



Figure 1

directory is `$JAVA_HOME/jre/lib/server/`.

Note: For Mac users, try the `bsd-libhsdis-i386.dylib` file and rename it to `hsdis-amd64.dylib` before copying it to the Java library directory.

The instructions [here](#) describe how to build your own HSDIS plugin from scratch.

Many Java developers have only limited experience working with assembly code, so the prospect of using HSDIS seems daunting. However, with time and some diligence, a working knowledge of assembly code is well within the grasp of the majority of developers. Here are some pointers to help

newcomers get comfortable working with assembly code:

- Don't panic. Assembly code isn't the end of the world.
- Get a good reference (online resource or a textbook).
- Understand the difference between different syntaxes for writing down assembly code. By default, HSDIS uses AT&T syntax, but it can be made to use Intel syntax instead.
- If you're familiar with C programming, remember that C compilers can be instructed not to emit machine code but to instead stop at the assembly code stage (for example, by using this com-

mand: `gcc -S`). Doing this can provide assembly code that corresponds to simple examples, which can aid understanding for beginners.

- Remember that Java HotSpot VM is a complicated runtime that has to deal with both compiled and interpreted methods. It also heavily optimizes compiled code and needs to cater to speculative optimizations that might need to be backed out. It's OK not to understand the reason for everything that's going on in an assembly code dump that came from Java HotSpot VM. Let's create a log that includes

Open Log in the main window to load in the test log.

JITWatch prompts you to click **Start** to process and parse the log, so do that next. You should end up with a screen similar to **Figure 3** that shows, in the left panel, a view of packages containing compiled methods.

If we drill into the packages, we can identify individual classes and methods and see how they've been treated by the JIT compiler, as shown in **Figure 4**.

There are also context-sensitive menus that work directly from the results page. See **Figure 5**.

We can even drill in to particular compilation trees by selecting the **Show compile chain** option, which leads to a screen that is similar to **Figure 6**.

Note: Tooltips available in the compile chain view show more detail about how the compiler handled methods. For example, in **Figure 6**, the tooltip indicates that the compiler compiled the method, but the method was too big to inline.

We're now ready to start working with the log in detail, so let's examine the "triview" screen next.

JITWatch Triview Screen

The triview is one of the most useful JITWatch screens. It shows three different views of the same

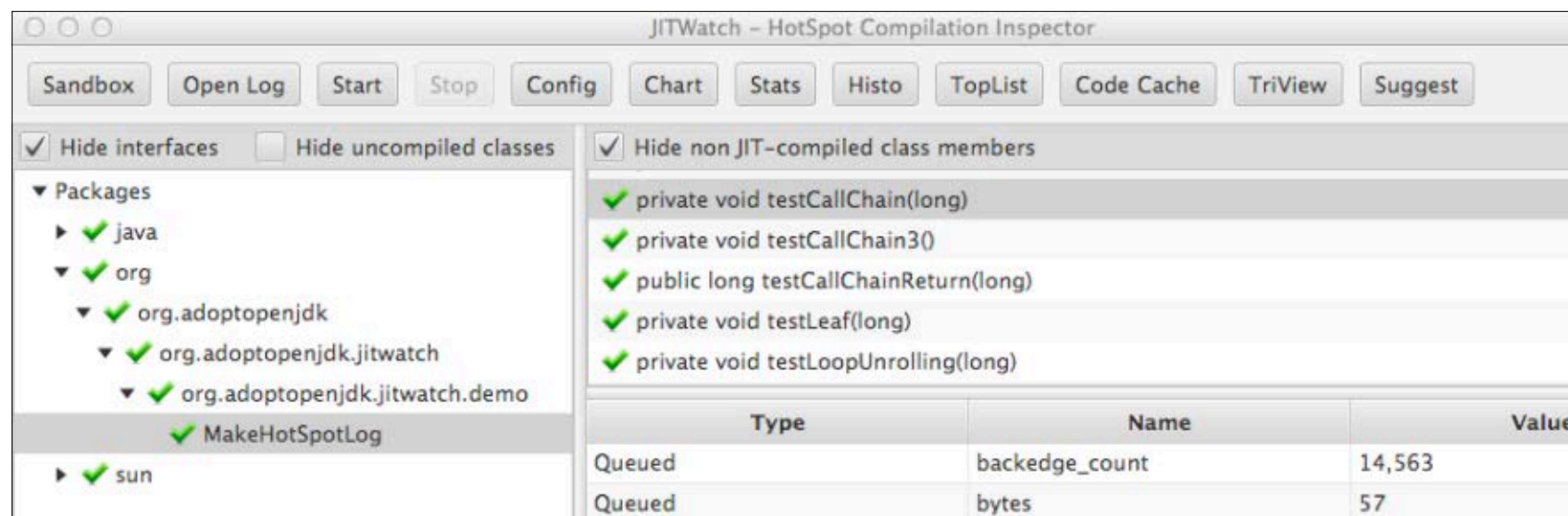


Figure 4

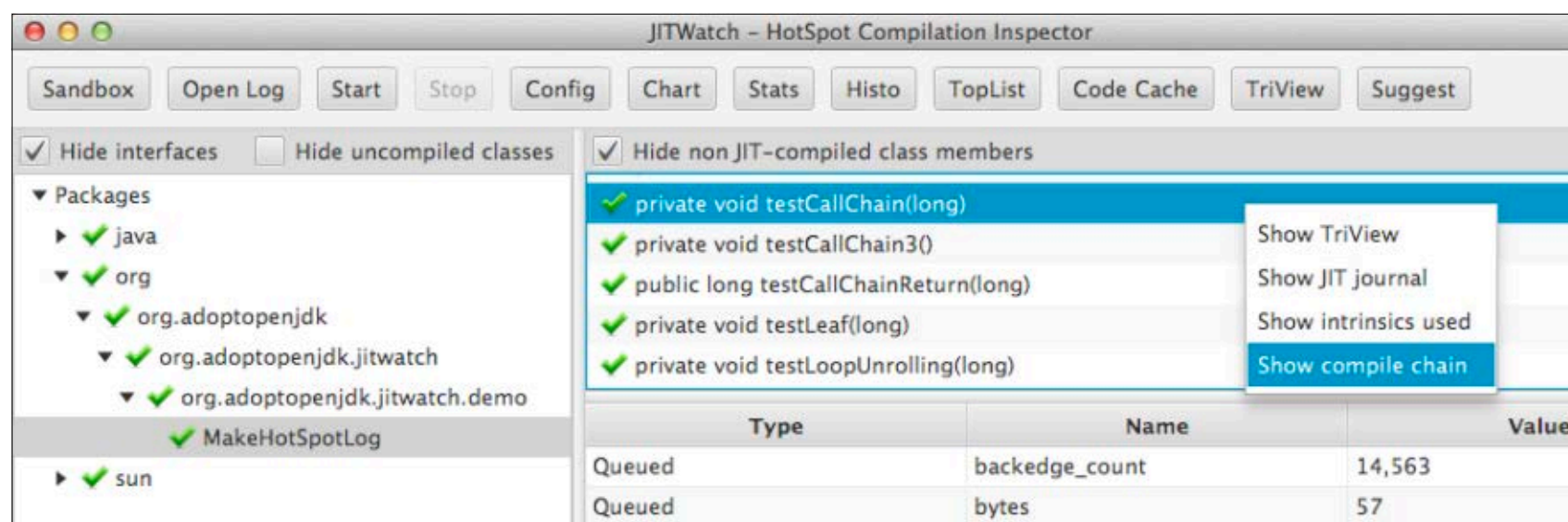


Figure 5

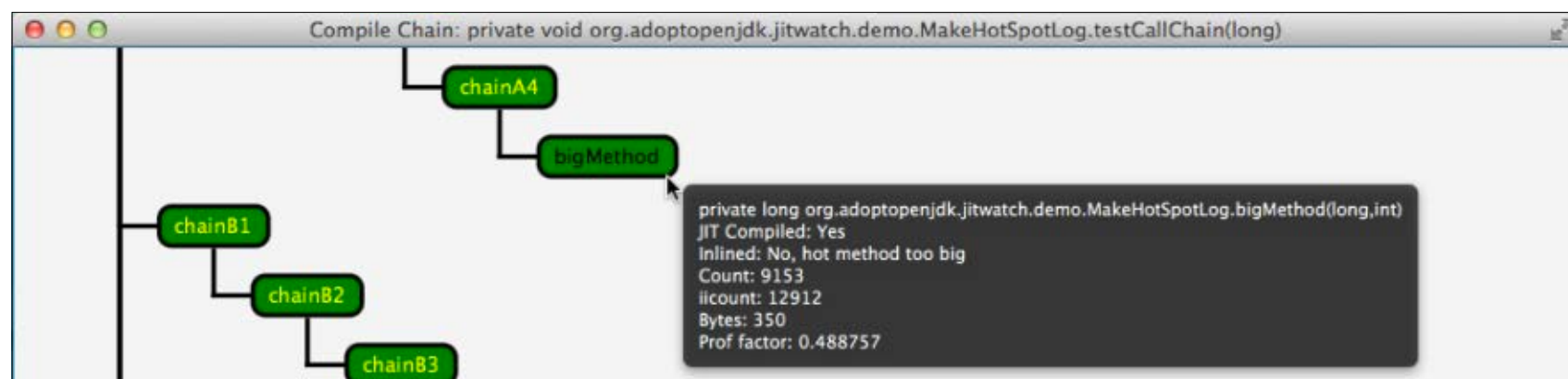


Figure 6

and add them to the list of components you can use later on.

Past experience with spaghetti code-based coding (servlets, JavaServer Pages) taught us about the importance of proper functional layer separation when building and maintaining apps, and in Oracle Mobile Application Framework, this is implemented with the MVC design pattern.

For the controller layer, Oracle Mobile Application Framework provides a process flow engine that allows you to define navigation between pages in your applications and also include method calls, decision points, and other flows in a flow. Code in the controller layer is written in simple Java classes that are similar in concept to managed beans in JSF.

Managed beans can also contain data that will be exposed in the UI layer. Oracle Mobile Application Framework provides various scopes for beans to better handle memory allocation. In addition, data can be incorporated from a remote server through both REST (JSON or XML) and SOAP services.

Oracle Mobile Application Framework uses a declarative data-binding layer to reduce the amount of code you need to manually write to hook up your UI components to the data and business services. At development time, a simple drag

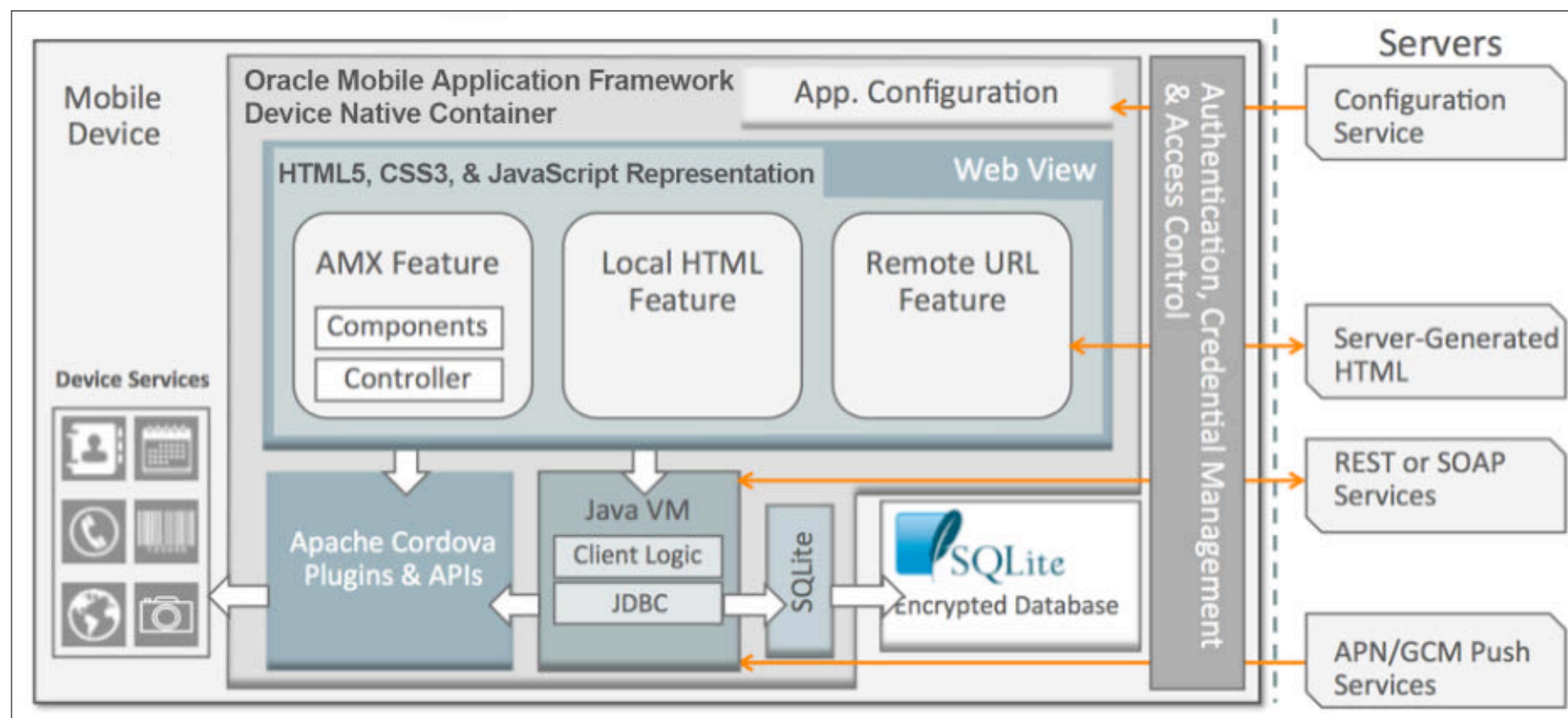


Figure 2

and drop will create the expression language that will connect a UI component to a datasource.

Building Truly Mobile Applications

A truly mobile application leverages the platform it runs on. For example, a truly mobile expense app lets you take pictures of receipts with the device's camera, a truly mobile product catalog uses GPS on the device to show the nearest store where you can buy an item, and a truly mobile HR system lets you quickly add an employee you are looking at to the contact list on your phone.

Oracle Mobile Application Framework lets you build these types of truly mobile apps by exposing device features for easy integration into your application, as shown in **Figure 2**. Using the open source Apache Cordova solution, Oracle Mobile Application Framework provides Java and JavaScript APIs that let you invoke the camera, GPS, SMS, e-mail, and other services directly from your code.

Furthermore, the framework supports the Cordova plugin architecture to let you add other device-specific capabilities to your app for features such as barcode scanning and temperature measurement.

Another aspect of mobile apps that Oracle Mobile Application Framework can leverage is push notification. Leveraging either the Apple or Google push server, a mobile app can receive push notifications and react to them with simple Java listeners that you code.

Oracle Mobile Application Framework further supports offline scenarios for mobile apps with a built-in encrypted SQLite database that apps can use to store data. Access to files on the local device's file system is also supported from within Oracle Mobile Application Framework.

With all of these capabilities at

MANAGE YOUR CAREER PATH AT DICE.COM

Java Jobs at your Fingertips

*Dice is the leading career site for
information technology and
engineering professionals.*

312,503
Java Jobs



*Register, upload your resume and
browse our wide range of tech and
IT positions.*

Dice®

priceline.com®

"For us, Java has become a way of life"

- Mike Diliberto, priceline.com CIO



We live Java. 24/7.

We're Hiring

Priceline.com is looking for Java Developers.
Apply today at: www.priceline.com/careers

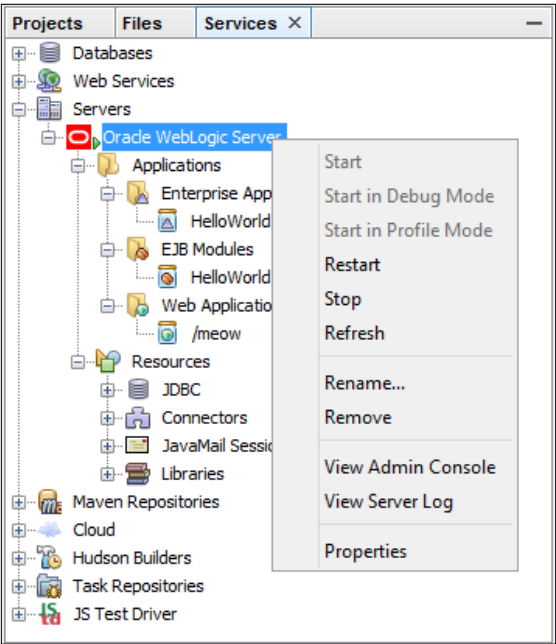


Figure 2

Now that we have quickly explored these tools, we are ready to dive into the building blocks of Oracle WebLogic Server.

Oracle WebLogic Server Components

A *domain* is the basic administration unit for Oracle WebLogic Server. This administration unit consists of one or more Oracle WebLogic Server instances and is managed by one instance that has a special role—the so-called Administration Server (AdminServer for short).
The AdminServer hosts the admin console and manages zero or more Oracle WebLogic Server instances called *managed servers*, which can be hosted locally to the AdminServer or hosted remotely

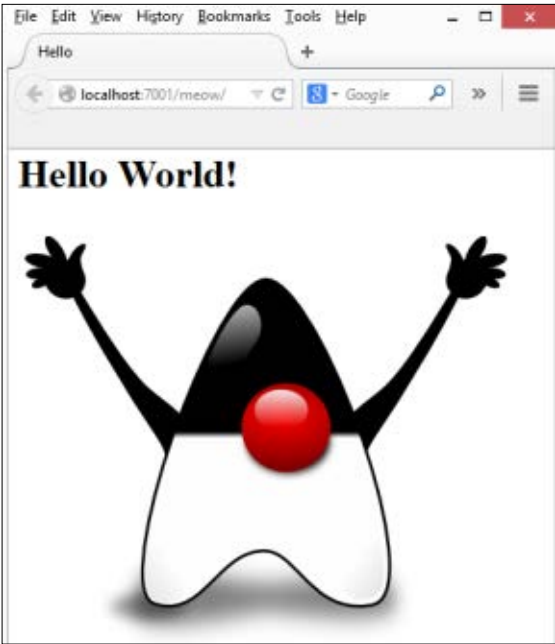


Figure 3

on different physical machines. Managed servers can be grouped into clusters. If managed servers are located on remote machines, separate NodeManager instances can be used for lifecycle operations, such as starting and stopping managed servers.
The central configuration file is called `config.xml`. This file is stored on the AdminServer with other files, such as database configuration files and security files. Each AdminServer manages exactly one domain, and all changes or activities apply only to this domain. Every domain can have multiple clusters. Every cluster can—but does not need to—include managed servers hosted on different machines.
In a nutshell, we have the Oracle WebLogic Server components

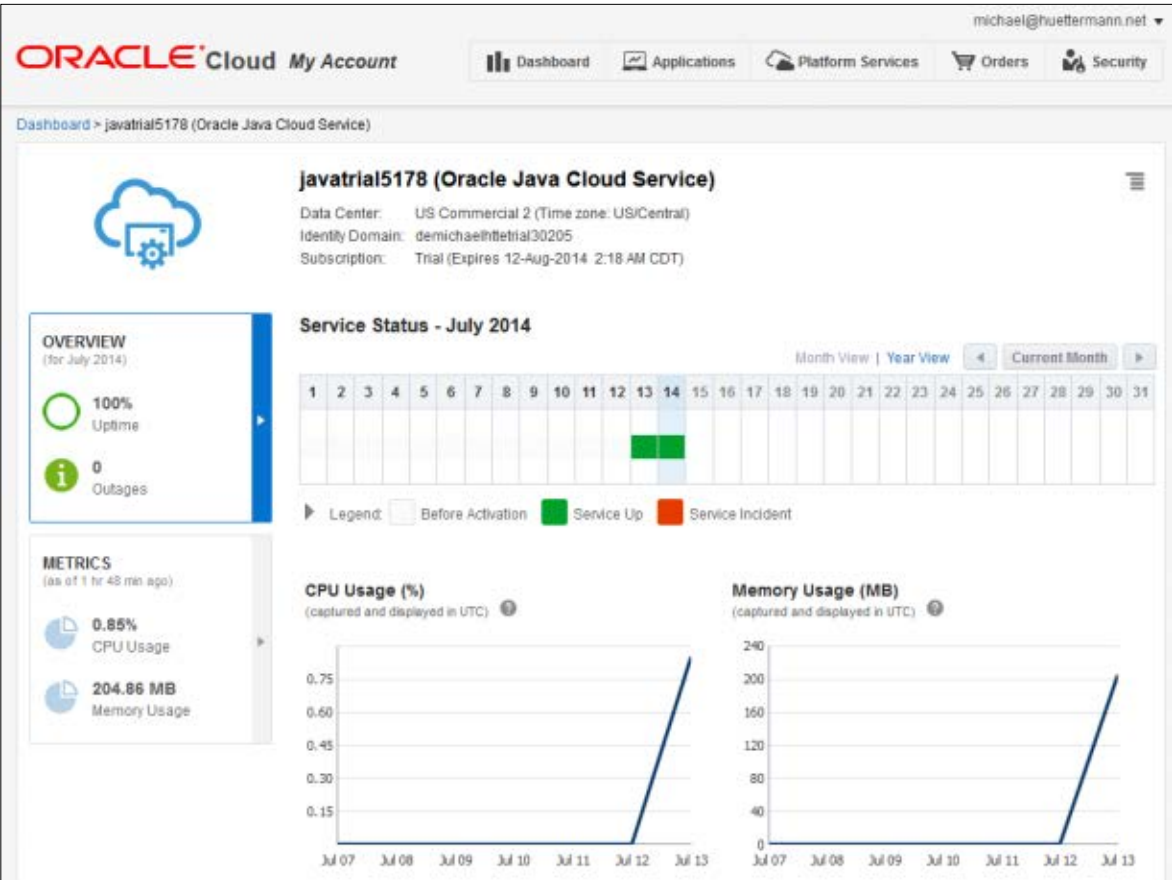


Figure 4

COMPONENT	DESCRIPTION
ADMINSERVER	A DOMAIN INCLUDES ONE ORACLE WEBLOGIC SERVER INSTANCE THAT IS CONFIGURED AS THE ADMINSERVER. ALL CHANGES TO THE CONFIGURATION AND DEPLOYMENT OF APPLICATIONS ARE DONE THROUGH THE ADMINSERVER.
MANAGED SERVERS	ALL OTHER ORACLE WEBLOGIC SERVER INSTANCES IN A DOMAIN ARE CALLED MANAGED SERVERS. MANAGED SERVERS HOST APPLICATION COMPONENTS AND RESOURCES, WHICH ARE ALSO DEPLOYED AND MANAGED AS PART OF THE DOMAIN.
CLUSTERS	A DOMAIN CAN ALSO INCLUDE ORACLE WEBLOGIC SERVER CLUSTERS, WHICH ARE GROUPS OF MANAGED SERVER INSTANCES THAT WORK TOGETHER TO PROVIDE SCALABILITY AND HIGH AVAILABILITY FOR APPLICATIONS. CLUSTERS CAN IMPROVE PERFORMANCE AND PROVIDE FAILOVER WHEN A SERVER INSTANCE BECOMES UNAVAILABLE.

Table 2

shown in **Table 2**.
In production environments, it is highly recommended that you deploy applications only on managed servers and you reserve the AdminServer for per-

forming management tasks. For more information about these basic components, see **Figure 5**, which shows example content of an Oracle WebLogic Server domain, and [“WebLogic Server Domains.”](#)

Let's now discuss clustered Oracle WebLogic Server instances and how to get your applications ready for enterprise usage.

Using Clustered Instances of Oracle WebLogic Server

Clustering applications makes them enterprise-ready. With clustering, incoming requests can be routed to an Oracle WebLogic Server instance in the cluster based on the volume of work being processed. In case of hardware failure or other failures, session-state information is available to other cluster nodes that can resume the work of the failed node. In addition, you can implement clusters so that services can be hosted on a single machine with the option to migrate the services to another node in the event of failure.

A clustered application or application component is one that is available on multiple Oracle WebLogic Server instances in a cluster. The following types of objects can be clustered in an Oracle WebLogic Server deployment. If an object is clustered,

failover and load balancing are available for that object.

- Servlets
- JSPs
- EJBs
- RMI objects
- JMS destinations
- Oracle Coherence clusters and managed Oracle Coherence servers
- Timer services

Clustered Oracle WebLogic Server instances behave similarly to nonclustered instances, except that they provide failover and load balancing. The process and tools used to configure clustered Oracle WebLogic Server instances are the same as those used to configure nonclustered instances.

Clustering provides enterprise-level benefits, above all the following:

- **Scalability.** The capacity of an application deployed on an Oracle WebLogic Server cluster can be increased dynamically to meet demand. You can add Oracle WebLogic Server instances to a cluster without interrupting the application's services.

Clusters consist of server instances that can be dynamically scaled up to meet the resource needs of your applications.

- High availability (HA). In an Oracle WebLogic Server clus-

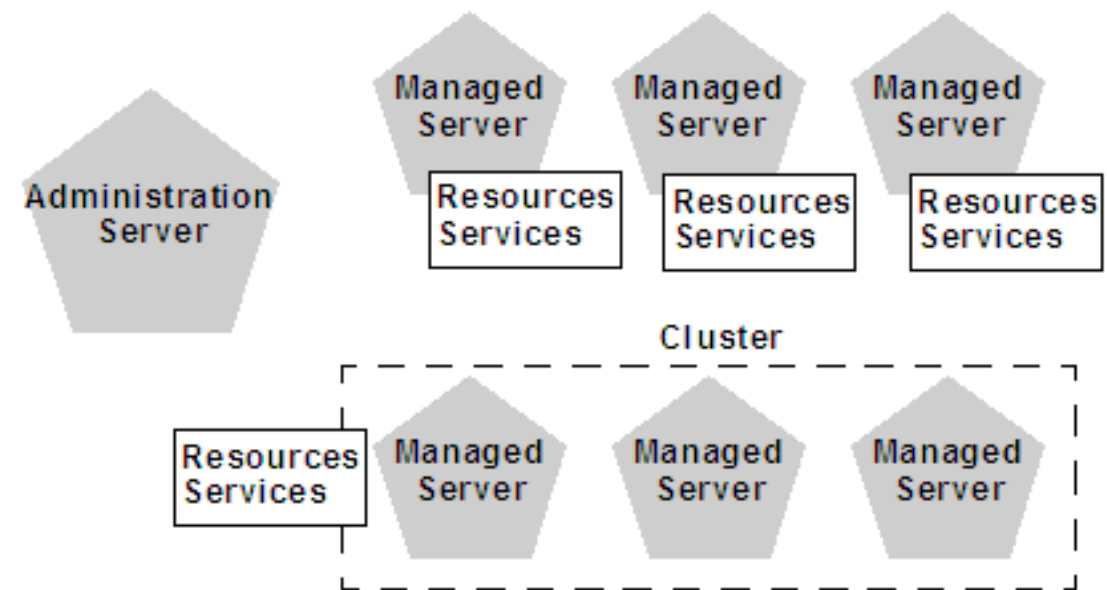


Figure 5

ter, application processing can continue when a server instance fails. You “cluster” application components by deploying them on multiple server instances in the cluster—so, if a server instance on which a component is running fails, another server instance on which that component is deployed can continue application processing.

Key clustering capabilities that enable scalability and HA include the following:

- Application failover. *Failover* means that when an application component doing a particular task becomes unavailable for any reason, a copy of the failed object finishes the task.
- Migration. Oracle WebLogic Server supports automatic and manual migration of a clus-

tered server instance from one machine to another. A managed server that can be migrated is referred to as a *migratable server*. This feature is designed for environments with HA requirements.

- Load balancing. *Load balancing* is the even distribution of jobs and associated communication across the computing and networking resources in your environment.

An example of how to achieve HA and scalability is by using a JDBC GridLink datasource in your application with Oracle WebLogic Server, which we discuss in the next section.

Achieving Persistence

With Oracle WebLogic Server, you have a couple of options for connecting to a DBMS. There are two



Discover the benefits of using JavaFX for visualization software.

The OpenMapFX project is an open source, community-driven project that provides a number of location-based services and tools. The core

In this article, we see

Figure 1 shows the only tile at zoom level 0 returned by the OpenStreetMap tile provider. This image can be obtained

Each incremental zoom level contains four times the number of tiles as the previous zoom level. For each tile



Figure 1

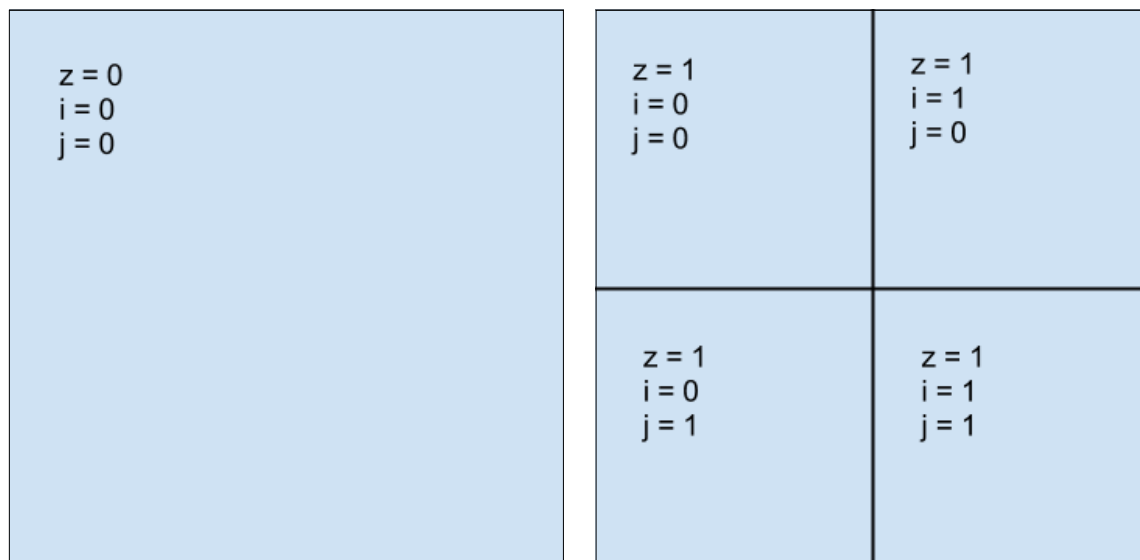


Figure 2



Figure 3

at zoom level z , there exist four tiles at zoom level $z + 1$ that cover the same area in more detail. **Figure 2** shows how the single tile at zoom level 1 corresponds to the four tiles at zoom level 2.

Rendering a single tile in JavaFX is very straightforward. We create

an `Image` object and pass the URL to the constructor. Next, we wrap the `Image` object in an `ImageView` instance, which can be rendered in the scene. The code in **Listing 1** shows how to do this.

The result of running the code shown in **Listing 1** is a 256 by 256 image that shows a static world map (see **Figure 3**).

Dragging and Scaling a Map

As a first enhancement, we will make our map draggable.

The JavaFX platform provides a number of features to deal with drag events and, by extension, with drag-and-drop events. We will take a simple approach here and listen for `mousePressed` and `mouseDragged` events and move the `imageView` accordingly. The code in **Listing 2** shows how to do this.

LISTING 1 LISTING 2 LISTING 3

```
@Override
public void start(Stage primaryStage) {
    Image image = new Image(
"http://tile.openstreetmap.org/0/0/0.png");
    ImageView imageView = new ImageView(image);
    StackPane root = new StackPane();
    root.getChildren().add(imageView);
    Scene scene = new Scene(root,300, 300);
    primaryStage.setTitle("TileMap 1");
    primaryStage.setScene(scene);
    primaryStage.show();
}
```

 [Download all listings in this issue as text](#)

We define the `onMousePressed` and `onMouseDragged` handlers on the root of the scene. In the `onMouseDragged` handler, we calculate the distance and change the `translate` property of the `imageView` node.

The map is now draggable, and we can move it inside the application window. As a next step, we want to be able to scale the map to make it bigger. Depending on the platform, different approaches for scaling are available. On a system with a mouse that has a scroll-wheel, moving the scrollwheel often indicates the desire to zoom in or out. In JavaFX, this behavior can easily be detected, and an appropriate handler can be used

to scale the image. **Listing 3**, which is taken from the TileMap3 source code sample, shows how to do this.

In **Listing 3**, we introduce a `DoubleProperty` named `scaleProperty`, which holds the scale factor of our map. We use the JavaFX binding concept to bind the scale of the node containing the image to the value of the `scaleProperty`. Rather than have the event handlers directly change the scale of the node, we let the event handlers change the `scaleProperty`. One advantage of using this approach is that we can have a number of event handlers—or pieces of code, in general—that modify the `scaleProperty` and, hence, cause the scale of the node to change.

string that is used to identify and authorize your openHAB instance. Note that the string should be at least 10 characters long.

For the HTTP binding, we need to configure the [http:weather.url](http://weather.url) property, which holds the URL for the location from which data should be retrieved. We will use Yahoo weather for San Francisco, so set the parameter to <http://weather.yahooapis.com/forecastrss?w=2487956&u=f>.

Additionally, set `http:weather.updateInterval` to `60000`, which means the URL is refreshed once every minute. **Listing 1** summarizes the content of our resulting `openhab.cfg` file.

Setting up a domain model. Once the general configuration is done, we can start designing the world inside openHAB.

Many home automation systems reflect the real devices in as detailed a manner as possible in the software. At first sight, this seems to be the most natural approach. However, every new device type has to be supported individually, which means it has to be implemented in the system before it can be used by the users. Sometimes, device hierarchies help mitigate this problem, but there are still compromises.

Rather than modeling a tumble dryer or a radio as a device, open-

HAB follows a capability-based approach. It splits devices into their functionalities, which are dubbed *items* in openHAB. This way, a radio can be modeled as three items: a switch representing the power, a dimmer for the volume, and a number specifying the station pre-set frequency.

Items can receive commands and hold a state. In order to keep the domain model compact, there are only a limited number of item types, for example, **String**, **Number**, **Dimmer**, **Contact**, **Rollershutter**, and **Color**.

Items are defined in files with the extension “.items” and they are stored in the folder `configurations/`[`items`](#). First, create a new items file called `my.items` in this folder. Once the file is created, it can be edited by the Designer.

Let's now create an item for a light that is capable of changing color based on the temperature in San Francisco. We use the type **Color** and give the item the unique name **Light**. To model a temperature, we use the type **Number** and the name **Temperature**. Since openHAB cannot infer an appropriate icon from the generic **Number** type, we specify a default icon **<temperature>** (or **<wind>** for the wind speed). Using parentheses an item can be assigned to a group, which is itself defined as an

LISTING 1

LISTING 2

```
#####
#####          General configurations          #####
#####

security:option=ON

##### HTTP Binding #####
# configuration of the first cache item
http:weather.url
    http://weather.yahooapis.com/forecastrss?w=2487956&u=f
http:weather.updateInterval=60000

##### Philips Hue Binding #####
hue:ip=192.168.0.42
hue:secret=myPersonalSecret
```

 [Download all listings in this issue as text](#)

item of type `Group`. The resulting `my.items` file is shown in **Listing 2**.

Next up is the user interface. Instead of offering a visual designer, openHAB follows a declarative UI approach. Users merely define the content to be shown on a certain page. The various openHAB UI implementations take care of an appropriate layout and the rendering themselves. These declarations are done in files called *sitemaps*. With a few lines, it is possible to define the structure and the content of the screens. Sitemap files are stored in the folder [configurations/sitemaps](#).

Each sitemap has an ID and a label (for example, "JavaMag") and

consists of lists of user interface elements for the content. (A list of all available UI elements can be found in the [GitHub wiki](#).) Screens can be portioned using the **Frame** sections.

Our **Light** item can be represented using the **Colorpicker** element with a slider icon. And our **Temperature** item can be represented using a **Text** element with a label that contains additional formatting options for the item's state. These formatting options allow for specifying the decimal fraction or adding a unit description. **Figure 1** shows the resulting sitemap for these items.

Binding the domain model to the world. Now that we have created

our domain model, we need to bind the items to the real world. This is done by adding a binding configuration to the item definitions. For the Philips Hue bulb, `{ hue="1" }` is added to the `Color` item, where `1` specifies the number of the bulb in the Philips Hue system (which has multiple bulbs) that we want to control.

The HTTP binding uses a slightly more complex syntax, but in essence, we need to reference the URL that we set up in `openhdb.cfg` and provide a suitable transformation function that extracts the piece of data that we require. This is done using XSLT stylesheets, which are readily prepared in the `configurations/transform` folder. The final `my.items` file is shown in **Listing 3**.

Storing and visualizing data. For all use cases where data needs to be permanently stored, openHAB comes with built-in persistence support, which is not restricted to a single datastore. Multiple

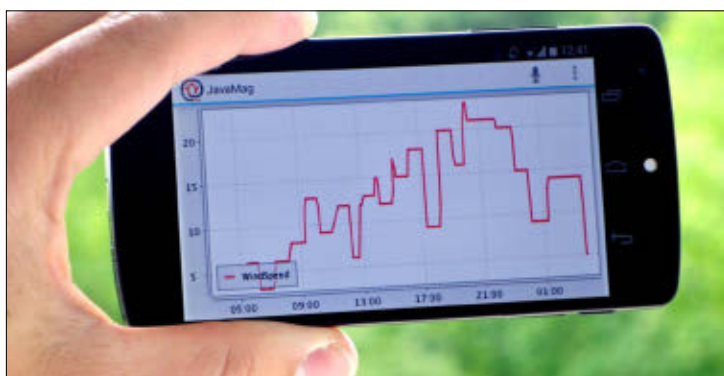


Figure 2

stores—such as relational databases, NoSQL databases, round-robin databases, Internet of Things (IoT) cloud services, and simple log files—can coexist and can be configured independently. Note that some options might be suitable only for exporting data (for example, IoT services or log files), while others can be queried as well, so they can be used for visualizing data through charts.

For every persistence service, a configuration file named `<persistence-service>.persist` (for example, `rrd4j.persist`) is placed in the folder `configurations/persistence`. These configuration files provide a simple way to define different time-based or event-based persistence individually for items.

For our example, **Listing 4** shows how data for all items belonging to the group **Temperatures** is stored in a round-robin database (RRD) every minute. In addition to storing item states, the persistence service also restores their states at system startup.


Once the data is stored to a queryable persistence service, the built-in chart engine can generate charts to visualize the data. There is a dedicated **Chart** element that can be used within sitemaps. **Figure 2** shows the result of ren-

LISTING 3 / LISTING 4 / LISTING 5

```

Color Light { hue="1" }
Number Temperature <temperature> (Temperatures) \
{ http="<[weather:60000:XSLT(yahoo_weather_temperature.xml)]" }
Number Forecast <temperature> (Temperatures) \
{ http="<[weather:60000:XSLT(yahoo_weather_forecast_high.xml)]" }
Number WindSpeed <wind> \
{ http="<[weather:60000:XSLT(yahoo_weather_wind_speed.xml)]" }
Group Temperatures

```

 [Download all listings in this issue as text](#)

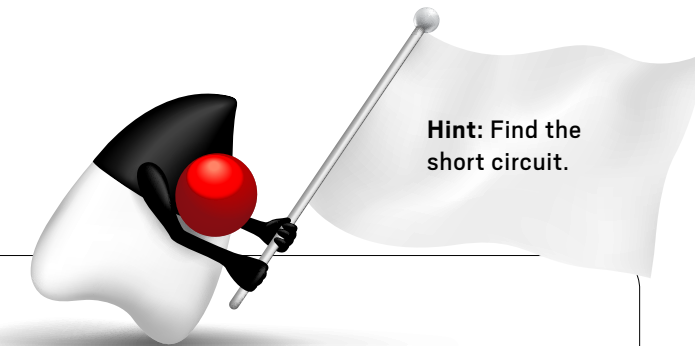
dering data as charts in the user interface.

Automating your world. Most of the fun use cases for a smart home are related to automation. In order to allow complex as well as unusual use cases, openHAB comes with a textual rule language. A *rule* in openHAB consists, in general, of one or many triggering conditions and a code block. The triggering conditions can be either event-based or time-based.

If multiple triggers are listed for a rule, any of them will trigger the code execution. The syntax of the code block of a rule uses a custom script language that is similar to Java and makes it easy to directly refer to items and their states.

Let's create the rule for our example use case. The triggering condition should be the change of the **Forecast** item's status, which is simply formulated as **Item Forecast changed**. In the code block, we have to transform the temperature value into a hue value (which represents a degree on the Hue's color circle). Let's map 60°F to 240° (=blue) on the hue scale and map 80°F to 0° (=red).

All that is left to do is to send a command that determines the appropriate color to show for the projected forecast encoded as HSB (hue, saturation, brightness) to the item `Light`. The resulting rule is shown in **Listing 5**. We store this rule for weather forecast visualiza-



In the July/August 2014 issue,

Abhishek Gupta asked what happens when we mix generic and raw types. The correct answer is #3. Compilation is successful but there is a `java.lang.ClassCastException` at runtime. This exception is due to the fact that a `java.lang.Integer` type makes its way into the `List` because it is not parameterized, and hence the compiler fails to apply type safety. It is fixed by changing the signature of the `add` method to use generics.

Answer #1 is incorrect because adding a primitive datatype (for example, `int`) is perfectly legal.

Answer #2 is incorrect because the program compiles successfully since compile type safety cannot be applied to a raw `List` type.

Answer #4 is incorrect. The program does generate a `java.lang.ClassCastException`, but usage of `List<Object>` as the method parameter does not work because generic types are invariant.

This issue's code teaser comes from Cyril Lapinte, a senior Java consultant in Geneva, Switzerland, who presents us with a streams challenge.

1 THE PROBLEM

The Java SE 8 `java.util.stream` package is a powerful toolset providing many optimizations for manipulating data within collections. The complexity is hidden, letting developers concentrate

on the "what," not on the "how."

Nonetheless, developers should still keep in mind the underlying mechanisms.

2 THE CODE

This program finds the `COUNT` number of prime numbers that are greater than some random starting value. It runs slowly because it's not making effective use of the Stream API. What change could be made that would result in a considerable speedup?

```
static final int MAXSEEDVALUE = 200_000;
static final int SEEDVALUE = new Random().nextInt(MAXSEEDVALUE);
static final int COUNT = 10;

System.out.println(
    IntStream.rangeClosed(SEEDVALUE + 1, MAXSEEDVALUE)
        .parallel()
        .filter(i -> IntStream.range(2, i)
            .filter(j -> i % j == 0)
            .count() == 0)
        .limit(COUNT)
        .mapToObj(String::valueOf)
        .collect(Collectors.joining(" ")));
```

3 WHAT'S THE FIX?

- 1) Replace the lambda expression `(j -> i % j == 0)` by an anonymous class implementing the `IntPredicate` interface.
- 2) Move the `limit()` function before the outer `filter()`.
- 3) Use `noneMatch()` instead of the inner `filter()`.
- 4) Replace both `filter()` functions using the iteration of the Java Collections Framework.

GOT THE ANSWER?

Look for the answer in the next issue. Or submit your own code challenge!